Reward Only Training of Encoder-Decoder Digit Recognition Systems Based on Policy Gradient Methods

Yilong Peng, Hayato Shibata, Takahiro Shinozaki
Tokyo Institute of Technology
www.ts.ip.titech.ac.jp

***Abstract*—Zero resource speech recognition is getting attention for engineering as well as scientific purposes. Based on the existing unsupervised learning frameworks using only speech input, however, it is impossible to associate automatically found linguistic units with spellings and concepts. In this paper, we propose an approach that uses a scalar reward that is assumed to be given for each decoding result of an utterance. While the approach is straightforward using reinforcement learning, the difficulty is to obtain a convergence without the help of supervised learning. Focusing on encoder-decoder based speech recognition, we explore several neural network architectures, optimization methods, and reward definitions, seeking a suitable configuration for policy gradient reinforcement learning. Experiments were performed using connected digit utterances from the TIDIGITS corpus as training and evaluation sets. While it is challenging, we show that learning a connected digit recognition system is possible achieving 13.6% of digit error rate. The success largely depends on the configurations and we reveal the appropriate condition that is largely different from supervised training.**

I. Introduction

Automatic speech recognition systems have reached human performance for several tasks [1], [2]. However, the performance largely depends on supervised learning, and there is a significant gap between human and speech recognition systems regarding the learning ability. The vulnerability in the learning results in substantial development cost that limits the application of speech recognition technologies to only a few major languages leaving most of the others in the world. To address the problem, and to answer the fundamental scientific question how a system can autonomously acquire language, speech recognition research in zero resource scenario is getting attention [3] where zero resource refers to no orthographic transcript. However, existing unsupervised subword modeling and spoken term discovery methods can not learn spelling and meaning of an utterance from interactions with humans. To realize such ability, reinforcement learning of speech recognition system would be needed, which is the focus of this paper.

Reinforcement learning is popular in spoken dialogue systems to improve dialogue control [4], [5], [6]. On the other hand, studies to apply it to learn speech models were limited [7] until recently. For an end-to-end speech recognition system based on connectionist temporal classification (CTC) [8], Graves firstly used expected transcription loss [9] which is equivalent to the REINFORCE policy gradient method [10] using the word error rate as the negative reward [11]. Since then, policy gradient methods have been used in CTC [12], [13] and encoder-decoder [14] based speech recognition systems to reduce word error rate by directly minimizing it rather than using surrogate objectives such as cross-entropy. Another application is a model adaptation based on user feedback available in cloud services [15]. In all of these systems, the policy gradient method is used to fine-tune systems that are initialized by supervised learning.

To the best of our knowledge, there has been no research that successfully applied reinforcement learning to speech recognition system in reward only training scenario without the supervised initialization. This is partly because of its difficulty. When the speech recognition system is randomly initialized, the recognition result is just random at the beginning of the learning both in the length of the hypothesis and its contents. Since the reward is given based on the system output, the learning becomes very difficult if most of the recognition results are completely wrong as mentioned in [9].

In this paper, we explore several neural network architectures, optimization methods, and reward definitions, seeking a suitable configuration for the policy gradient reinforcement learning focusing on encoder-decoder based speech recognition systems. The assumptions are that a system gets scalar feedback for its output per utterance and no other information is given. Specifically, we compare Likelihood Ratio Method (LRM) [16], [17] and Proximal Policy Optimization (PPO) algorithm [18] as the variants of the policy gradient methods and combine them with several definitions of rewards based on word error rate. As the network architectures, we investigate LSTM based basic encoder-decoder networks [19], [20] and their extensions. While the learning is challenging, we show that it is possible to train a connected digit recognition system based on an encoder-decoder network from scratch using only scalar reward by optimizing the network structure and the learning configuration.

II. Policy Gradient Methods

Policy gradient methods are types of reinforcement learning algorithms. As the general setup, a system has a set of actions and a policy function f that takes a state or observation $\boldsymbol{s}$ and returns a probability distribution $\pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})$ of an action a to take. The policy function is parameterized by a set of parameters $\boldsymbol{\theta}$. From $\pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})$, an action is sampled and

executed. In our case, $\boldsymbol{s}$ is an acoustic feature sequence, and the action is a recognition hypothesis. According to the action, the system gets a scalar reward $r_{\boldsymbol{s}}(a)$.

A. Likelihood ratio method (LRM)

The goal of the learning is to maximize the expected reward $\mathbb{E}[r_{\boldsymbol{s}}(a)] = \sum_a \pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})\, r_{\boldsymbol{s}}(a)$ with respect to $\boldsymbol{\theta}$. The maximization can be performed by applying the gradient ascent method. However, there may not exist an analytic functional form of the reward, and enumerating all possible actions may not be tractable. Therefore, LRM uses log derivative trick or likelihood ratio $\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|\boldsymbol{s}) = \frac{\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})}{\pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})}$ as in the natural evolution strategy [21], [22].

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathbb{E}[r_{\boldsymbol{s}}(a)|\boldsymbol{\theta}] &= \nabla_{\boldsymbol{\theta}} \sum_a \pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})\, r_{\boldsymbol{s}}(a) \\ &= \sum_a \pi_{\boldsymbol{\theta}}(a|\boldsymbol{s}) \left(\frac{\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})}{\pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})} \right) r_{\boldsymbol{s}}(a) \\ &= \mathbb{E}[r_{\boldsymbol{s}}(a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})]. \end{aligned} \quad (1)$$

Equation (1) means that $r_{\boldsymbol{s}}(a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})$ is an unbiased estimator of the gradient $\nabla_{\boldsymbol{\theta}} \mathbb{E}[r_{\boldsymbol{s}}(a)|\boldsymbol{\theta}]$ that can be evaluated by sampling action a from $\pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})$ and obtaining its reward. Given the estimate of the gradient, the parameter update formula is obtained as follows.

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta} + \epsilon r_{\boldsymbol{s}}(a) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|\boldsymbol{s}), \quad (2)$$

where ϵ (> 0) is the learning rate. This algorithm is called REINFORCE if the policy function is a neural network. When a toolbox is used that supports automatic differentiation to implement the neural network, the parameter update can be conducted by using Equation (3) as the error function of the network negating the sign of the objective.

$$-r_{\boldsymbol{s}}(a) \log \pi_{\boldsymbol{\theta}}(a|\boldsymbol{s}). \quad (3)$$

If the reward is constant, Equation (3) becomes equivalent to the cross-entropy error.

B. Proximal policy optimization (PPO)

Several extensions of LRM have been proposed. Among them, Trust Region Policy Optimization (TRPO) has demonstrated good performance in benchmarking [23]. PPO is an efficient extension of TRPO that is simple to implement. Compared to LRM, it minimizes Equation (4) instead of Equation (3), where $\pi_{\boldsymbol{\theta}_{old}}$ is the policy function with old parameters, δ is a non-negative value to specify a clipping threshold, and $clip$ is a clipping function that clips the value in the first argument between the range specified by the second and the third arguments.

$$\begin{aligned} \max \quad & \left\{ -\frac{\pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})}{\pi_{\boldsymbol{\theta}_{old}}(a|\boldsymbol{s})} r_s, \right. \\ & \left. -\mathrm{clip}\left(\frac{\pi_{\boldsymbol{\theta}}(a|\boldsymbol{s})}{\pi_{\boldsymbol{\theta}_{old}}(a|\boldsymbol{s})}, 1-\delta, 1+\delta \right) r_s \right\}. \end{aligned} \quad (4)$$

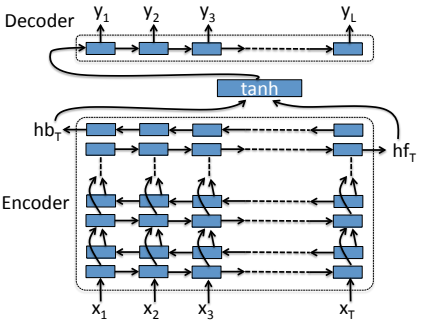

Fig. 1. Linear model. The input $x_1, x_2, \cdots, x_T$ is an acoustic feature sequence with variable length T, and the output $y_1, y_2, \cdots, y_L$ is a digit sequence with variable length L.

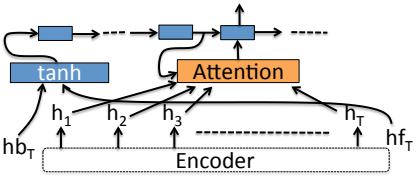

Fig. 2. Attention model.

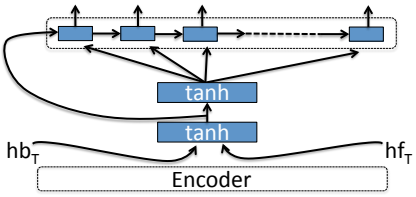

Fig. 3. Spoke(out) model.

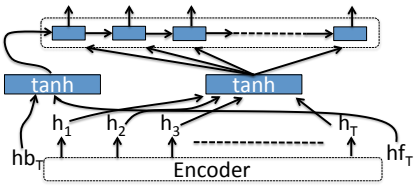

Fig. 4. Spoke(in,out) model.

III. Encoder-Decoder Models

For the connected digit recognition, we investigate four encoder-decoder models with different architectures in the reward only training scenario. The first model is a simple network having 5-layer BiLSTM with 128 forward and 128 backward units as the encoder network and 1-layer 256 units LSTM as the decoder network as shown in Figure 1. The two outputs of the encoder network of the opposite directions are concatenated and 256-dimensional utterance vector is made through a $tanh$ layer. The structure is similar to the Sutskever's machine translation model [19]. We refer to this model as "**linear**" in the followings. The second model integrates an attention mechanism [24] to the first model as shown in Figure 2, which we refer to as "**attention**". The third model

is similar to the Cho's encoder-decoder machine translation model [20] having radiating spokes from a $tanh$ layer located on top of the utterance vector to each frame of the decoder network. The encoder and the decoder are the same as the first model, and the hub-like additional $tanh$ layer has 512 units. We refer to this model as "**spoke(out)**". The last model is similar to the third model but we extend it by adding spokes from the encoder network to the $tanh$ layer. We refer to this model as "**spoke(in,out)**".

In all the networks, the output layer of the decoder network is a softmax having 12 units representing digits (0 to 9 and "o") and an end-of-string <EOS> symbol. We do not make a link to connect $(l-1)$-th output to l-th input in the decoder network since the task is digit recognition and no correlation is assumed between the adjacent digits. In fact, there was no big difference in the performance when we introduced the link in our preliminary experiments.

IV. Rewards

For speech recognition, accuracy is the most basic evaluation measure. As an initial work on the reward only training, it would be the first choice as a measure to simulate human feedback focusing on how to use the given reward.

In our preliminary experiments, we have found that there is a problem when we directly used it as the reward for a digit sequence drawn from the posterior distribution by the decoder network. That is, because the decoder output is random at the beginning and the accuracy is very low, the systems tend to stick to 1-length output just to prevent insertion errors. Based on this observation, we define and investigate the following rewards.

A. Clipped accuracy (ClpAcc)

As mentioned, accuracy (Acc) would be the first choice for the reward. It is defined for each utternace by Equation (5), where N_{ref} it the number of digits in a reference, E is the number of recognition errors, and Err is the error rate. The number of errors E is a sum of the numbers of insertion (I), deletion (D), and substitution (D) errors. To use Acc as a reward, we define ClpAcc as shown in Equation (6) clipping negative value to 0 since otherwise the learning has diverged in our preliminary experiments.

$$Acc = \frac{N_{ref} - E}{N_{ref}} = 1 - Err, \quad (5)$$

$$ClpAcc = \max\{Acc, 0\}. \quad (6)$$

B. Symmetric accuracy (SymAcc)

To prevent the recognition system from sticking to the 1-length output, we define SymAcc as shown in Equation (7), where $N_{hyp} = N_{ref} + I - D$ is the length of the hypothesis. As shown in Equation (8), SymAcc discounts the accuracy when the length of the hypothesis is shorter than the reference, which motivates the system to try longer hypothesis.

$$SymAcc = \max\left\{\frac{N_{ref} - E}{2N_{ref}} + \frac{N_{hyp} - E}{2N_{hyp}}, 0\right\} \quad (7)$$

$$= \max\left\{1 - \frac{1 - Acc}{2}\left(1 + \frac{N_{ref}}{N_{hyp}}\right), 0\right\}. \quad (8)$$

C. Length error penalized accuracy (LPAcc)

Another strategy to prevent the system to persist on the 1-length output is to explicitly penalize the difference between the reference and the hypothesis lengths. We define LPAcc as shown in Equation 9, where α is a penalty coefficient. In the experiment, α was chosen to 0.3 based on a preliminary experiment.

$$LPAcc = \max\{Acc - \alpha\,|N_{ref} - N_{hyp}|, 0\}. \quad (9)$$

D. SymAcc with reward mean clipping (SymAcc+RMC)

It is known that sometimes it is useful to consider relatively whether the reward is better or worse compared to the current expectation to reduce estimation variance, and reinforcement baseline (also referred to as reward baseline) is used [25], [26]. Here, we evaluate a strategy to clip the reward by its mean in the preceding samples as shown in Equation (10). In the following experiment, we combined it with SymAcc.

$$SymAccRMC = \begin{cases} SymAcc & (SymAcc \geq m) \\ 0 & (otherwise) \end{cases}, \quad (10)$$

where m is the mean of the original accuracy Acc in the previous 8.5k samples. At the beginning of the learning, we set $m = 0$ as an initial condition. The meaning is that we only use samples that outperform the standard level at that timing.

V. Experimental setup

Experiments were performed using adult English speech data of connected digits from the TIDIGITS corpus [1]. The training data included 4.2 hours of spoken utterances from 55 male and 57 female speakers, which consisted of 8623 samples of utterances of 1 to 7 digits. The test data included 4.3 hours of utterances from 56 male and 57 female speakers. Table I shows the distribution of the utterance length in the training set. Acoustic features were 13-dimensional MFCC extracted using Kaldi toolkit with the default settings specifying the sampling rate to 20kHz. In the reinforcement learning, we draw training samples from the training set with replacement. The information that was given to the system was only the reward that was evaluated for the recognition hypothesis. The hypothesis was obtained by sampling from the estimated posterior distribution by the system for a given input. For the implementation, Tensorflow [27] was used. The mini-batch size was set to 64.

TABLE I
Distribution of utterance length in the training set.

# digits	1	2	3	4	5	6	7
frequency	2464	1232	1232	1332	1132	0	1231

[1] https://catalog.ldc.upenn.edu/ldc93s10

TABLE II
DIGIT ERROR RATE (DER) OF THE ENCODER-DECODER MODELS WHEN THEY WERE TRAINING BY SUPERVISED LEARNING.

	linear	attention	spoke(out)	spoke(in,out)
DER	21.5%	1.3%	16.1%	6.7%

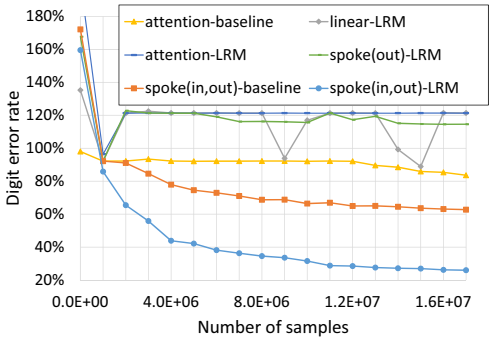

Fig. 5. Test set digit error rate (DER) by the encoder-decoder models when they were trained by reinforcement learning from scratch based on the SymAcc reward. The horizontal axis is the number of processed training samples.

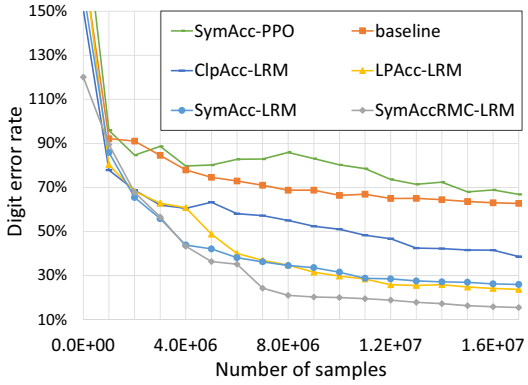

Fig. 6. Test set digit error rate (DER) by the spoke(in, out) model comparing the combinations of the rewards and the optimization algorithms, and the baseline.

As a baseline to compare, we evaluated a strategy of performing supervised learning gathering samples whose labels were correctly identified just by chance by a random guess. The success rate of guessing the correct label varies according to how to randomly guess the label. If we consider obtaining a distribution q_i by which the chance of guessing a correct label $\sum_{i=1}^{N} p_i q_i$ is maximized assuming a prior distribution of the labels p_i is known, where i is an index of a distinct label, it is a problem of linear programming. The answer is to set $q_k = 1$ where $k = \underset{i}{\operatorname{argmax}}\, p_i$ and $q_j = 0$ for all $j \neq k$. In this case, the training is performed using only the samples with the most frequent labels, which is apparently not desirable. Here, we chose $q_i = p_i$ as the baseline strategy, where the chance of getting a correct label of a digit sequence was 0.77% for a sample in the training set.

VI. RESULTS

Table II shows test set digit error rate (DER) of the encoder-decoder models when they were trained by supervised learning[2]. For the learning rate control, ADAM [28] was used. The attention model gave lower DER than the linear model as expected. The DER of the Spoke(in, out) model was higher than the attention model but lower than the linear model.

Figure 5 shows the results of reward only training using the SymAcc reward and the LRM learning algorithm comparing different model architectures. Different from the supervised learning, we used SGD since ADAM did not work at all. The learning rate was set to 0.0005 based on a preliminary experiment and it is kept constant. When the baseline learning strategy was applied to the attention model, the system always outputs the identical single digit at the beginning and the learning hardly proceeded. After iterating more than $1.2 * 10^7$ times, it began to produce different digits and the DER reduced a little, but they were still a single digit. This was probably partly because the selected training samples were biased to 1 digit length utterances since the chance of guessing correct label p_i^2 was high (See Table I for their distribution), and also partly because of the network structure. The attention mechanism may have effect of making the network stick on the wrong alignment. When the baseline strategy was used for the spoke(in, out) model, the DER gradually reduced.

When the reinforcement learnings were performed using the linear, attention and spoke(out) models, the learning did not proceed, keeping DER mostly in the region over 100%. However, spoke(in, out) model worked well. The DER monotonically reduced, and DER of 26.0% was obtained. The advantages of spoke(in, out) are that it does not consider alignment as the attention mechanism and robust for the wrong alignment, and yet can convey more information than the linear model through the spoke connection from the encoder network to the decoder network.

To verify the assumption about the problem of the attention mechanism, we first applied a few epochs of supervised learning and then performed the reinforcement learning as an additional experiment. Since the attention becomes accurate, the learning should progress. When starting from DER of 12.0%, we have confirmed that reduced WER of 10.8% was obtained by the LRM training using adjusted learning rate of 10^{-7} as expected. The result is also consistent with that of previous researches that applied reinforcement learning in the supervised learning scenario for the fine tunings purpose.

Figure 6 compares DER by the spoke(in, out) model when different reward types and the optimization algorithms are combined. The DER based on the baseline learning is also shown. It is seen that LRM gave better results than PPO in this task. All the learning using LRM gave better results than the baseline. LPAcc and SymAcc gave similar performance that was better than ClpAcc when LRM was used. The best result was obtained when SymAccRMC was used with LRM. After processing $3.1 * 10^7$ samples by LRM with SymAccRMC the DER become 13.6% demonstrating the possibility of the

[2]The definition of DER is the same as word error rate (WER) but we use the term DER since the unit is a digit

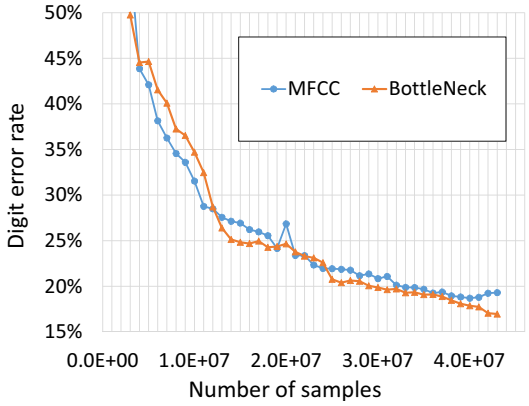

Fig. 7. Digit error rate (DER) when features developed for the ZeroSpeech2017 were used.

reward only learning of encoder-decoder recognition systems.

Finally, we performed the reward only training using features developed for the ZeroSpeech2017 challenge [29]. They were 40-dimensional bottleneck features where the feature extraction neural network was trained using a spontaneous Japanese speech corpus Corpus of Spontaneous Japanese [30]. For the cross-lingual evaluation based on ABX-test in the challenge, it largely reduced the error rate, especially in across-speaker condition. Figure 7 shows the results when the features were used with spoke(in, out), LRM, and SymAcc. In the figure, "MFCC" is the MFCC features same as other experiments, and "BottleNeck" is the bottleneck features. While the difference was small, the bottleneck feature gave better performance than the MFCC features. When BottleNeck was used, the DER was 14.41% after processing $6.5 * 10^7$ samples.

VII. Conclusions

We have proposed scalar reward only training approach for zero resource speech recognition, and have investigated several encoder-decoder neural network architectures, optimization methods, and reward definitions, seeking a suitable configuration for policy gradient reinforcement learning. We have found that there is a tendency in the initial learning process that the system persists on 1-length output as a strategy to avoid insertion errors when it is initialized randomly. To avoid the problem, we have proposed SymAcc and LPAcc rewards. We have also found that the attention mechanism prevents the progress of the learning when the learning starts from scratch, and our proposed spoke(in, out) structure works better in such situation avoiding the problem. The best result was obtained when spoke(in, out) model was trained with LRM and SymAccRMC, which introduced the reward mean clipping to the SymAcc reward. The lowest DER was 13.6%. Future work includes reducing the required number of training samples by combining unsupervised learning, and extending the task from digit recognition to language understanding utilizing more general reward.

VIII. Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 26280055 and 17K20001.

References

[1] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," in *Proc. Interspeech 2017*, 2017, pp. 132–136.

[2] A. Stolcke and J. Droppo, "Comparing human and machine errors in conversational speech transcription," in *Proc. Interspeech*. ISCA - International Speech Communication Association, August 2017, pp. 137–141.

[3] M. Versteegh, R. Thiollière, T. Schatz, X. N. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015," in *Proc. Interspeech*, 2015, p. 3169 3173.

[4] D. Lu, T. Nishimoto, and N. Minematsu, "Decision of response timing for incremental speech recognition with reinforcement learning," in *2011 IEEE Workshop on Automatic Speech Recognition Understanding*, Dec 2011, pp. 467–472.

[5] P. H. Su, D. Vandyke, M. Gasic, D. Kim, N. Mrksic, T. H. Wen, and S. J. Young, "Learning from real users: rating dialogue success with neural networks for reinforcement learning in spoken dialogue systems." in *Proc. Interspeech*, 2015, pp. 2007–2011.

[6] F. Wang, "A multi-agent reinforcement learning algorithm for disambiguation in a spoken dialogue system," in *Proceedings of the 2010 International Conference on Technologies and Applications of Artificial Intelligence*, ser. TAAI '10. IEEE Computer Society, 2010, pp. 116–123.

[7] C. Molina, N. B. Yoma, F. Huenupan, C. Garreton, and J. Wuth, "Maximum entropy-based reinforcement learning using a confidence measure in speech recognition for telephone speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 1041–1052, July 2010.

[8] A. Graves, S. Fernndez, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *In Proceedings of the International Conference on Machine Learning, ICML 2006*, 2006, pp. 369–376.

[9] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.

[10] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229–256, May 1992.

[11] D. Bahdanau, D.Serdyuk, P. Brakel, N. R. Ke, J. Chorowski, A. Courville, and Y. Bengio, "Task loss estimation for sequence prediction," in *International Conference on Learning Representation*, 2016, pp. 1–13.

[12] M. Shannon, "Optimizing expected word error rate via sampling for speech recognition," in *Proc. Interspeech 2017*, 2017, pp. 3537–3541.

[13] Y. Zhou, C. Xiong, and R. Socher, "Improving end-to-end speech recognition with policy learning," *arXiv preprint arXiv:1712.07101*, 2017.

[14] A. Tjandra, S. Sakti, and S. Nakamura, "Sequence-to-sequence asr optimization via reinforcement learning," *arXiv preprint arXiv:1710.10774*, 2017.

[15] T. Kato and T. Shinozaki, "Reinforcement learning of speech recognition system based on policy gradient and hypothesis selection," in *Proc. ICASSP*, 2018, accepted.

[16] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, ser. NIPS'99, 1999, pp. 1057–1063.

[17] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. PMLR, 2016, pp. 1928–1937.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14, 2014, pp. 3104–3112.
[20] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP*, 2014, pp. 1724–1734.
[21] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, "Natural evolution strategies," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 949–980, 2014.
[22] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.
[23] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning*, 2016, pp. 1329–1338.
[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
[25] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," in *Reinforcement Learning*. Springer, 1992, pp. 5–32.
[26] L. Weaver and N. Tao, "The optimal reward baseline for gradient-based reinforcement learning," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 538–545.
[27] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'16. Berkeley, CA, USA: USENIX Association, 2016, pp. 265–283.
[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[29] H. Shibata, T. Kato, T. Shinozaki, and S. Watanabe, "Composite embedding system for zerospeech 2017 track1," in *Proc. ASRU*, 2017, pp. 747–753.
[30] S. Furui, K. Maekawa, and H. Isahara, "A Japanese national project on spontaneous speech corpus and processing technology," in *Proc. ASR'00*, 2000, pp. 244–248.